





```
1 0001 0 MODULE lib_compress ( ! Compress the library
2 0002 0 LANGUAGE (BLISS32),
3 0003 0 IDENT = 'V04-000'
4 0004 0 ) =
5 0005 1 BEGIN
6 0006 1
7 0007 1
8 0008 1 *****
9 0009 1 *
10 0010 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
11 0011 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
12 0012 1 * ALL RIGHTS RESERVED.
13 0013 1 *
14 0014 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
15 0015 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
16 0016 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
17 0017 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
18 0018 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
19 0019 1 * TRANSFERRED.
20 0020 1 *
21 0021 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
22 0022 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
23 0023 1 * CORPORATION.
24 0024 1 *
25 0025 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
26 0026 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
27 0027 1 *
28 0028 1 *
29 0029 1 *****
30 0030 1
31 0031 1 ++
32 0032 1
33 0033 1 FACILITY: Library command processor
34 0034 1
35 0035 1 ABSTRACT:
36 0036 1
37 0037 1 The VAX/VMS librarian is invoked by DCL to process the LIBRARY
38 0038 1 command. It utilizes the librarian procedure set to perform
39 0039 1 the actual modifications to the library.
40 0040 1
41 0041 1 ENVIRONMENT:
42 0042 1
43 0043 1 VAX native, user mode.
44 0044 1
45 0045 1 --
46 0046 1
47 0047 1
48 0048 1 AUTHOR: Benn Schreiber, CREATION DATE: 22-June-1979
49 0049 1
50 0050 1 MODIFIED BY:
51 0051 1
52 0052 1 V03-003 MCN0145 Maria del C. Nasr 08-Feb-1984
53 0053 1 When using the /COMPRESS qualifier, if the input library
54 0054 1 is in data reduced format, do not expand the new one.
55 0055 1 Expansion will only be done when /DATA=EXPAND is used.
56 0056 1
57 0057 1 V03-002 GJA0064 Greg Awdziewicz 26-Jan-1984
```



:	58	0058	1	:		Allow benign compression of an empty library.
:	59	0059	1	:		
:	60	0060	1	:	V03-001	JWT0056 Jim Teague 16-Sep-1982
:	61	0061	1	:		Equipped with DCX interface for /COMPRESS=REDUCE.
:	62	0062	1	:		
:	63	0063	1	:	V02-007	RPG0037 Bob Grosso 15-Jan-1982
:	64	0064	1	:		Use library history attributes rather than default.
:	65	0065	1	:		
:	66	0066	1	:	V02-006	RPG0036 Bob Grosso 18-Dec-1981
:	67	0067	1	:		Improve error reporting with update history
:	68	0068	1	:		
:	69	0069	1	:	V02-005	RPG0035 Bob Grosso 7-Aug-1981
:	70	0070	1	:		lib\$gl_ctlmsk now a quadword.
:	71	0071	1	:		
:	72	0072	1	:	V02-004	RPG0034 Bob Grosso 30-Jul-1981
:	73	0073	1	:		Support CREATE=KEEP.
:	74	0074	1	:		
:	75	0075	1	:	V02-003	BLS0029 Benn Schreiber 23-Dec-1980
:	76	0076	1	:		Change messages to use message compiler.
:	77	0077	1	:		
:	78	0078	1	:	V02-002	RPG0004 Bob Grosso 3-Sep-1980
:	79	0079	1	:		Exit read or write loops and print end of module header
:	80	0080	1	:		and continue compressing.
:	81	0081	1	:		
:	82	0082	1	:		
:	83	0083	1	:		



```
85 0084 1 LIBRARY
86 0085 1 'SYS$LIBRARY:STARLET.L32';
87 0086 1 REQUIRE
88 0087 1 'PREFIX';
89 0271 1 REQUIRE
90 0272 1 'LIBDEF';
91 0560 1 REQUIRE
92 0561 1 'LBRDEF';
93 1152 1
94 1153 1 EXTERNAL ROUTINE
95 1154 1   lib_get_mem,
96 1155 1   lbr$dcx_map: ADDRESSING_MODE (GENERAL),
97 1156 1   lbr$get_history : ADDRESSING_MODE (GENERAL), !Get the library history
98 1157 1   lbr$put_history : ADDRESSING_MODE (GENERAL), !Replace history
99 1158 1   lbr$get_index : ADDRESSING_MODE (GENERAL), !Call routine for index entries
100 1159 1   lbr$find : ADDRESSING_MODE (GENERAL), !Find module by RFA
101 1160 1   lbr$lookup_key : ADDRESSING_MODE (GENERAL), !Lookup key in index
102 1161 1   lbr$insert_key : ADDRESSING_MODE (GENERAL), !Insert new key into index
103 1162 1   lbr$put_end : ADDRESSING_MODE (GENERAL), !Terminate writing module text
104 1163 1   lbr$set_index : ADDRESSING_MODE (GENERAL), !Set index number to use
105 1164 1   lbr$set_module : ADDRESSING_MODE (GENERAL), !Read/update module header
106 1165 1   lbr$get_record : ADDRESSING_MODE (GENERAL), !Read text record
107 1166 1   lbr$put_record : ADDRESSING_MODE (GENERAL), !Write text record
108 1167 1   lbr$search : ADDRESSING_MODE (GENERAL), !Search an index for an RFA
109 1168 1   lbr$open : ADDRESSING_MODE (GENERAL), !Open library
110 1169 1   lbr$close : ADDRESSING_MODE (GENERAL), !Close library
111 1170 1   lbr$ini_control : ADDRESSING_MODE (GENERAL), !Initialize control index
112 1171 1   lbr$insert_time : ADDRESSING_MODE (GENERAL), !Set module insert date/time
113 1172 1   lib_log_op, !Log operation
114 1173 1   lib_create_lib; !Create output library
115 1174 1
116 1175 1 EXTERNAL
117 1176 1   lbr$gl_control : REF BBLOCK ADDRESSING_MODE (GENERAL), !Librarian control table address
118 1177 1   lbr$gl_rmsstv : ADDRESSING_MODE (GENERAL), !RMS STV from librarian
119 1178 1   lib$gl_type, !Type of library
120 1179 1   lib$al_hdrlen : VECTOR [LONG], !Lengths of various module headers
121 1180 1   lib$al_ascbinf : VECTOR [LONG], !Key lengths
122 1181 1   lib$gl_keysize, !Max size of key in library
123 1182 1   lib$gl_libctl : BLOCK [2], !Input library control index
124 1183 1   lib$gl_libfdb : REF BBLOCK, !Pointer to library FDB
125 1184 1   lib$gl_outfdb : REF BBLOCK, !Pointer to output library FDB
126 1185 1   lib$gl_ctlmsk : BLOCK [1], !Librarian control flags
127 1186 1   lib$gl_cre8flags : BITVECTOR, !Compress option flags
128 1187 1   lib$gl_allgbls, !Number of globals to allocate in new library
129 1188 1   lib$gl_allmods, !Number of modules to allocate in new library
130 1189 1   lib$gl_allksz, !Size of keys in new library
131 1190 1   lib$gl_allhis, !Max number of history records in new library
132 1191 1   lib$gl_objgsdix, !Index number of object globals
133 1192 1   lib$gl_modnamix; !Index number of module names
134 1193 1
135 1194 1 EXTERNAL LITERAL
136 1195 1   lbr$_nulidx, ! Index is empty.
137 1196 1   lib$_emptylibrary, ! An empty library is to be
138 1197 1   ! compressed.
139 1198 1   lib$_cnvrting, !Converting info message
140 1199 1   lib$_histerr, !Error in update history
141 1200 1   lib$_indexerr, !Some strange index error
```

```
: 142      1201 1      lib$_initerr,      !Initialization error
: 143      1202 1      lib$_inserted,    !Module inserted
: 144      1203 1      lib$_inserterr,    !Error inserting into index
: 145      1204 1      lib$_lookuperr,    !Error looking up module
: 146      1205 1      lib$_mhderr;       !Module header error
: 147      1206 1
: 148      1207 1      FORWARD ROUTINE
: 149      1208 1      lib_put_history,    !Copy over the history records.
: 150      1209 1      get_index_if_not_empty, !Call Lbr$Get_index.
: 151      1210 1      copymodule,        !Copy one object module
: 152      1211 1      enterglobals;      !Enter globals for obj lib
: 153      1212 1
: 154      1213 1      GLOBAL
: 155      1214 1      dcx_map_desc : VECTOR [2];
: 156      1215 1
: 157      1216 1      OWN
: 158      1217 1      curindex,           !Current index being searched
: 159      1218 1      newtxtrfa : BBLOCK [dsc$_s_bln], !Module RFA in new library
: 160      1219 1      outlibindex,       !Control index for output library
: 161      1220 1      func : LONG INITIAL (lbr$_create); !Function to create library
```



```
163 1221 1 GLOBAL ROUTINE lib_comprs_lib (after_func) =
164 1222 2 BEGIN
165 1223 2 ++
166 1224 2 FUNCTIONAL DESCRIPTION:
167 1225 2
168 1226 2 This routine compresses one library into another.
169 1227 2
170 1228 2 CALLING SEQUENCE:
171 1229 2
172 1230 2 status = lib_comprs_lib (after_func)
173 1231 2
174 1232 2 INPUT PARAMETERS:
175 1233 2
176 1234 2 after_func is the function (lbr$read, lbr$update) to open the compressed library with
177 1235 2 after the compress has been completed
178 1236 2
179 1237 2 IMPLICIT INPUTS:
180 1238 2
181 1239 2 lib$gl_libfdb is the pointer to the library (input FDB)
182 1240 2 lib$gl_outfdb is the pointer to the output FDB
183 1241 2
184 1242 2 IMPLICIT OUTPUTS:
185 1243 2
186 1244 2 lib$gl_libfd is changed to point to the output FDB
187 1245 2
188 1246 2 SIDE EFFECTS:
189 1247 2 NONE
190 1248 2
191 1249 2 --
192 1250 2
193 1251 2 LOCAL
194 1252 2
195 1253 2 usrmohdrln, ! temp store expansion size of module header
196 1254 2 header : REF BBLOCK,
197 1255 2 status;
198 1256 2
199 1257 2 BIND
200 1258 2
201 1259 2 libdesc = lib$gl_libfdb [fdb$l_namdesc] : BBLOCK, !Name the filename descriptor
202 1260 2 outdesc = lib$gl_outfdb [fdb$l_namdesc] : BBLOCK, ! for input and output libraries
203 1261 2 libnamblk = lib$gl_libfdb [fdb$t_nam] : BBLOCK, !Name the NAM blocks
204 1262 2 outnamblk = lib$gl_outfdb [fdb$t_nam] : BBLOCK; ! ...
205 1263 2
206 1264 2
207 1265 2 Determine what create options we need to derive from input library
208 1266 2 and do it.
209 1267 2
210 1268 2 header = .lbr$gl_control [lbr$l_hdrptr]; !point to library header
211 1269 2 IF NOT .lib$gl_cre8flags [lib$opt_gbls] !Globals specified by option?
212 1270 2 THEN lib$gl_allgbls = .header [lhd$l_idxcnt] - .header [lhd$l_modcnt]; !No--compute from header
213 1271 2 IF NOT .lib$gl_cre8flags [lib$opt_mods] !Modules specified by option
214 1272 2 THEN lib$gl_allmods = .header [lhd$l_modcnt] + .header [lhd$l_idxovh];
215 1273 2 IF NOT .lib$gl_cre8flags [lib$opt_ksz] !Key size specified?
216 1274 2 THEN IF .lib$gl_ctlmsk [lib$v_o[dlb]]
217 1275 2 THEN lib$gl_allksz = .lib$al_ascbinf [.lib$gl_type] ! if old library, then get new size
218 1276 2 ELSE BEGIN
219 1277 2 !
```

```
220 1278 3 ! Get size of keys from input library if new format
221 1279 3 !
222 1280 3 BIND
223 1281 3 indexdesc = .header + lhd$c_idxdesc : BBLOCK; !Point to first index descriptor
224 1282 3
225 1283 3 lib$gl_allksz = .indexdesc [idd$w_keylen] - 1; !Get size of keys minus count byte
226 1284 3 END;
227 1285 3 lib$gl_keysize = .lib$gl_allksz; !Set key size for future reference
228 1286 3 lib$gl_cre8flags [lib$c_opt_ksz] = true; !Flag specified now
229 1287 3
230 1288 2 ! To determine the maximum number of history records for new library,
231 1289 2 ! if /COMPRESS=HISTORY:n specified then its value will be used,
232 1290 2 ! else use attribute from old library header.
233 1291 2
234 1292 2 IF NOT .lib$gl_cre8flags [lib$c_opt_luhs]
235 1293 2 THEN
236 1294 2 lib$gl_allhis = .header [lhd$w_maxluhrec];
237 P 1295 2 perform (lbr$ini_control (outlibindex, func, !Init the control index
238 P 1296 2 [lib$gl_type, outnamblk), lib$_initerr,
239 1297 2 1, outdesc);
240 1298 2
241 1299 2 ! If the user specified /COMPRESS, (not /DATA), and the input library
242 1300 2 ! is already reduced, keep it that way.
243 1301 2
244 1302 2 IF NOT .lib$gl_ctlmsk [lib$v_data]
245 1303 2 AND .header [lhd$l_dcxmapvbn] neq 0
246 1304 2 THEN
247 1305 2 lib$gl_cre8flags [lib$c_opt_dcx] = 1 ;
248 1306 2
249 1307 2 ! If we're creating a DCX-processed library...
250 1308 2
251 1309 2 IF .lib$gl_cre8flags [lib$c_opt_dcx]
252 1310 2 THEN
253 1311 2 perform ( lbr$dcx_map (lib$gl_libctl, dcx_map_desc ));
254 1312 2
255 1313 2 CH$MOVE (dsc$c_s_bln, lib$gl_libfdb [fdb$l_defext], !Set default ext.
256 1314 2 lib$gl_outfdb [fdb$l_defext]);
257 1315 2 outnamblk [nam$l_rlf] = libnamblk; !Set up related filename block
258 1316 2
259 1317 2 ! Save size of additional data area in module if /COMP=KEEP
260 1318 2
261 1319 2 usrmodhdrln = .lib$al_hdrln [.lib$gl_type]; !Save defaults
262 1320 2
263 1321 2 IF .lib$gl_ctlmsk [lib$v_keep]
264 1322 2 THEN
265 1323 2 lib$al_hdrln [.lib$gl_type] = .header [lhd$b_mhdusz]; !Use value in library
266 1324 2
267 1325 2 ! Create output library; make it with data reduced if it should be so.
268 1326 2
269 P 1327 2 perform (lib_create_lib (.lib$gl_outfdb, outlibindex,
270 P 1328 2 (IF .lib$gl_cre8flags [lib$c_opt_dcx] THEN dcx_map_desc
271 1329 2 ELSE 0) ));
272 1330 2 lib$al_hdrln [.lib$gl_type] = .usrmodhdrln; !Restore defaults
273 1331 2
274 1332 2 IF .lib$gl_ctlmsk [lib$v_convert] !If this is forced convert
275 1333 2 THEN SIGNAL (lib$_cnvrtng, 2, outdesc, libdesc); ! tell user whats happening
276 1334 2 !
```



```
277 1335 2 ! Call the library procedures to return each entry in the module name
278 1336 2 ! index. It will call copymodule for each entry.
279 1337 2
280 P 1338 2 rms_perform (get_index_if_not_empty(),
281 1339 2 [lib$indexerr, lbr$gl_rmsstv, 1, libdesc);
282 1340 2 IF .lib$gl_ctlmsk [lib$keep] !If history is to be retained then
283 1341 2 THEN
284 1342 2 BEGIN
285 1343 2 status = lbr$get_history (lib$gl_libctl, lib_put_history); !copy history
286 1344 2 IF NOT .status
287 1345 2 THEN
288 1346 2 SIGNAL (lib$hiterr, 1, libdesc, .status);
289 1347 2 END;
290 1348 2
291 P 1349 2 rms_perform (lbr$close (outlibindex), !Close the new library
292 1350 2 lib$closeout, .lbr$gl_rmsstv, 1, outdesc);
293 1351 2
294 P 1352 2 rms_perform (lbr$close (lib$gl_libctl), !and the old library
295 1353 2 lib$closein, .lbr$gl_rmsstv, 1, outdesc);
296 1354 2
297 1355 2 lib$gl_ctlmsk [lib$oldlib] = false; !No longer old library
298 1356 2 lib$gl_libfdb = .lib$gl_outfdb; !Make the library FDB
299 1357 2 ! the old output FDB
300 P 1358 2 perform (lbr$ini_control (lib$gl_libctl, after_func, !Init control block to open lib
301 1359 2 [lib$gl_type, outnamblk),
302 1360 2 lib$initerr, 1, outdesc);
303 1361 2
304 P 1362 2 rms_perform (lbr$open (lib$gl_libctl), !Open newly created library
305 1363 2 lib$openin, .lbr$gl_rmsstv, 1, outdesc);
306 1364 2
307 1365 2 RETURN true
308 1366 1 END;
```

!Of lib\_compress\_lib

```
.TITLE LIB_COMPRESS
.IDENT \V04-000\

.PSECT $OWNS$,NOEXE,2

00000 CURINDEX:
      .BLKB 4
00004 NEWTXTRFA:
      .BLKB 8
0000C OUTLIBINDEX:
      .BLKB 4
00000000 00010 FUNC: .LONG 0

.PSECT $GLOBAL$,NOEXE,2

00000 DCX_MAP_DESC::
      .BLKB 8

.EXTRN LIB_GET_MEM, LBR$DCX_MAP
.EXTRN LBR$GET_HISTORY
.EXTRN LBR$PUT_HISTORY
.EXTRN LBR$GET_INDEX, LBR$FIND
.EXTRN LBR$LOOKUP_KEY, LBR$INSERT_KEY
```

```
.EXTRN LBR$PUT_END, LBR$SET_INDEX
.EXTRN LBR$SET_MODULE, LBR$GET_RECORD
.EXTRN LBR$PUT_RECORD, LBR$SEARCH
.EXTRN LBR$OPEN, LBR$CLOSE
.EXTRN LBR$INI_CONTROL
.EXTRN LBR$INSERT_TIME
.EXTRN LIB_LOG_OP, LIB_CREATE_LIB
.EXTRN LBR$GL_CONTROL, LBR$GL_RMSSTV
.EXTRN LBR$GL_TYPE, LBR$GL_HDRLEN
.EXTRN LBR$GL_ASCBINF, LBR$GL_KEYSIZE
.EXTRN LBR$GL_LIBCTL, LBR$GL_LIBFDB
.EXTRN LBR$GL_OUTFDB, LBR$GL_CTLMSK
.EXTRN LBR$GL_CRE8FLAGS
.EXTRN LBR$GL_ALLGBLS, LBR$GL_ALLMODS
.EXTRN LBR$GL_ALLKSZ, LBR$GL_ALLHIS
.EXTRN LBR$GL_OBJGSDIX
.EXTRN LBR$GL_MODNAMIX
.EXTRN LBR$_NOLIDX, LBR$_EMPTYLIBRARY
.EXTRN LBR$_CNVRTING, LBR$_HISTERR
.EXTRN LBR$_INDEXERR, LBR$_INITERR
.EXTRN LBR$_INSERTED, LBR$_INSERTERR
.EXTRN LBR$_LOOKUPERR, LBR$_MHDERR

.PSECT $CODE$,NOWRT,2

.ENTRY LIB_COMPRS_LIB, Save R2,R3,R4,R5,R6,R7,R8,- ; 1221
R9,R10,R11
ADDL3 #16, LBR$GL_LIBFDB, R11 ; 1259
ADDL3 #16, LBR$GL_OUTFDB, R9 ; 1260
ADDL3 #64, LBR$GL_LIBFDB, R10 ; 1261
ADDL3 #64, LBR$GL_OUTFDB, R8 ; 1262
MOVL LBR$GL_CONTROL, R0 ; 1268
MOVL 10(R0), HEADER
BBS #1, LBR$GL_CRE8FLAGS, 1$ ; 1269
SUBL3 110(HEADER), 106(HEADER), LBR$GL_ALLGBLS ; 1270
BBS #2, LBR$GL_CRE8FLAGS, 2$ ; 1271
ADDL3 120(HEADER), 110(HEADER), LBR$GL_ALLMODS ; 1272
BBS #3, LBR$GL_CRE8FLAGS, 4$ ; 1273
TSTB LBR$GL_CTLMSK+2 ; 1274
BGEQ 3$ ; 1275
MOVL LBR$GL_TYPE, R0
MOVL LBR$GL_ASCBINF[R0], LBR$GL_ALLKSZ
BRB 4$ ; 1281
MOVAB 196(HEADER), R0 ; 1283
MOVZWL 2(R0), LBR$GL_ALLKSZ
DECL LBR$GL_ALLKSZ
MOVL LBR$GL_ALLKSZ, LBR$GL_KEYSIZE ; 1285
BISB2 #8, LBR$GL_CRE8FLAGS ; 1286
BBS #4, LBR$GL_CRE8FLAGS, 5$ ; 1292
MOVZWL 124(HEADER), LBR$GL_ALLHIS ; 1294
PUSHL R8 ; 1297
PUSHAB LBR$GL_TYPE
PUSHAB FUNC
PUSHAB OUTLIBINDEX
CALLS #4, LBR$INI_CONTROL
BLBS STATUS, 6$
PUSHL STATUS
```

Address	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418	Op419	Op420	Op421	Op422	Op423	Op424	Op425	Op426	Op427	Op428	Op429	Op430	Op431	Op432	Op433	Op434	Op435	Op436	Op437	Op438	Op439	Op440	Op441	Op442	Op443	Op444	Op445	Op446	Op447	Op448	Op449	Op450	Op451	Op452	Op453	Op454	Op455	Op456	Op457	Op458	Op459	Op460	Op461	Op462	Op463	Op464	Op465	Op466	Op467	Op468	Op469	Op470	Op471	Op472	Op473	Op474	Op475	Op476	Op477	Op478	Op479	Op480	Op481	Op482	Op483	Op484	Op485	Op486	Op487	Op488	Op489	Op490	Op491	Op492	Op493	Op494	Op495	Op496	Op497	Op498	Op499	Op500	Op501	Op502	Op503	Op504	Op505	Op506	Op507	Op508	Op509	Op510	Op511	Op512	Op513	Op514	Op515	Op516	Op517	Op518	Op519	Op520	Op521	Op522	Op523	Op524	Op525	Op526	Op527	Op528	Op529	Op530	Op531	Op532	Op533	Op534	Op535	Op536	Op537	Op538	Op539	Op540	Op541	Op542	Op543	Op544	Op545	Op546	Op547	Op548	Op549	Op550	Op551	Op552	Op553	Op554	Op555	Op556	Op557	Op558	Op559	Op560	Op561	Op562	Op563	Op564	Op565	Op566	Op567	Op568	Op569	Op570	Op571	Op572	Op573	Op574	Op575	Op576	Op577	Op578	Op579	Op580	Op581	Op582	Op583	Op584	Op585	Op586	Op587	Op588	Op589	Op590	Op591	Op592	Op593	Op594	Op595	Op596	Op597	Op598	Op599	Op600	Op601	Op602	Op603	Op604	Op605	Op606	Op607	Op608	Op609	Op610	Op611	Op612	Op613	Op614	Op615	Op616	Op617	Op618	Op619	Op620	Op621	Op622	Op623	Op624	Op625	Op626	Op627	Op628	Op629	Op630	Op631	Op632	Op633	Op634	Op635	Op636	Op637	Op638	Op639	Op640	Op641	Op642	Op643	Op644	Op645	Op646	Op647	Op648	Op649	Op650	Op651	Op652	Op653	Op654	Op655	Op656	Op657	Op658	Op659	Op660	Op661	Op662	Op663	Op664	Op665	Op666	Op667	Op668	Op669	Op670	Op671	Op672	Op673	Op674	Op675	Op676	Op677	Op678	Op679	Op680	Op681	Op682	Op683	Op684	Op685	Op686	Op687	Op688	Op689	Op690	Op691	Op692	Op693	Op694	Op695	Op696	Op697	Op698	Op699	Op700	Op701	Op702	Op703	Op704	Op705	Op706	Op707	Op708	Op709	Op710	Op711	Op712	Op713	Op714	Op715	Op716	Op717	Op718	Op719	Op720	Op721	Op722	Op723	Op724	Op725	Op726	Op727	Op728	Op729	Op730	Op731	Op732	Op733	Op734	Op735	Op736	Op737	Op738	Op739	Op740	Op741	Op742	Op743	Op744	Op745	Op746	Op747	Op748	Op749	Op750	Op751	Op752	Op753	Op754	Op755	Op756	Op757	Op758	Op759	Op760	Op761	Op762	Op763	Op764	Op765	Op766	Op767	Op768	Op769	Op770	Op771	Op772	Op773	Op774	Op775	Op776	Op777	Op778	Op779	Op780	Op781	Op782	Op783	Op784	Op785	Op786	Op787	Op788	Op789	Op790	Op791	Op792	Op793	Op794	Op795	Op796	Op797	Op798	Op799	Op800	Op801	Op802	Op803	Op804	Op805	Op806	Op807	Op808	Op809	Op810	Op811	Op812	Op813	Op814	Op815	Op816	Op817	Op818	Op819	Op820	Op821	Op822	Op823	Op824	Op825	Op826	Op827	Op828	Op829	Op830	Op831	Op832	Op833	Op834	Op835	Op836	Op837	Op838	Op839	Op840	Op841	Op842	Op843	Op844	Op845	Op846	Op847	Op848	Op849	Op850	Op851	Op852	Op853	Op854	Op855	Op856	Op857	Op858	Op859	Op860	Op861	Op862	Op863	Op864	Op865	Op866	Op867	Op868	Op869	Op870	Op871	Op872	Op873	Op874	Op875	Op876	Op877	Op878	Op879	Op880	Op881	Op882	Op883	Op884	Op885	Op886	Op887	Op888	Op889	Op890	Op891	Op892	Op893	Op894	Op895	Op896	Op897	Op898	Op899	Op900	Op901	Op902	Op903	Op904	Op905	Op906	Op907	Op908	Op909	Op910	Op911	Op912	Op913	Op914	Op915	Op916	Op917	Op918	Op919	Op920	Op921	Op922	Op923	Op924	Op925	Op926	Op927	Op928	Op929	Op930	Op931	Op932	Op933	Op934	Op935	Op936	Op937	Op938	Op939	Op940	Op941	Op942	Op943	Op944	Op945	Op946	Op947	Op948	Op949	Op950	Op951	Op952	Op953	Op954	Op955	Op956	Op957	Op958	Op959	Op960	Op961	Op962	Op963	Op964	Op965	Op966	Op967	Op968	Op969	Op970	Op971	Op972	Op973	Op974	Op975	Op976	Op977	Op978	Op979	Op980	Op981	Op982	Op983	Op984	Op985	Op986	Op987	Op988	Op989	Op990	Op991	Op992	Op993	Op994	Op995	Op996	Op997	Op998	Op999	Op1000	Op1001	Op1002	Op1003	Op1004	Op1005	Op1006	Op1007	Op1008	Op1009	Op1010	Op1011	Op1012	Op1013	Op1014	Op1015	Op1016	Op1017	Op1018	Op1019	Op1020	Op1021	Op1022	Op1023	Op1024	Op1025	Op1026	Op1027	Op1028	Op1029	Op1030	Op1031	Op1032	Op1033	Op1034	Op1035	Op1036	Op1037	Op1038	Op1039	Op1040	Op1041	Op1042	Op1043	Op1044	Op1045	Op1046	Op1047	Op1048	Op1049	Op1050	Op1051	Op1052	Op1053	Op1054	Op1055	Op1056	Op1057	Op1058	Op1059	Op1060	Op1061	Op1062	Op1063	Op1064	Op1065	Op1066	Op1067	Op1068	Op1069	Op1070	Op1071	Op1072	Op1073	Op1074	Op1075	Op1076	Op1077	Op1078	Op1079	Op1080	Op1081	Op1082	Op1083	Op1084	Op1085	Op1086	Op1087	Op1088	Op1089	Op1090	Op1091	Op1092	Op1093	Op1094	Op1095	Op1096	Op1097	Op1098	Op1099	Op1100	Op1101	Op1102	Op1103	Op1104	Op1105	Op1106	Op1107	Op1108	Op1109	Op1110	Op1111	Op1112	Op1113	Op1114	Op1115	Op1116	Op1117	Op1118	Op1119	Op1120	Op1121	Op1122	Op1123	Op1124	Op1125	Op1126	Op1127	Op1128	Op1129	Op1130	Op1131	Op1132	Op1133	Op1134	Op1135	Op1136	Op1137	Op1138	Op1139	Op1140	Op1141	Op1142	Op1143	Op1144	Op1145	Op1146	Op1147	Op1148	Op1149	Op1150	Op1151	Op1152	Op1153	Op1154	Op1155	Op1156	Op1157	Op1158	Op1159	Op1160	Op1161	Op1162	Op1163	Op1164	Op1165	Op1166	Op1167	Op1168	Op1169	Op1170	Op1171	Op1172	Op1173	Op1174	Op1175	Op1176	Op1177	Op1178	Op1179	Op1180	Op1181	Op1182	Op1183	Op1184	Op1185	Op1186	Op1187	Op1188	Op1189	Op1190	Op1191	Op1192	Op1193	Op1194	Op1195	Op1196	Op1197	Op1198	Op1199	Op1200	Op1201	Op1202	Op1203	Op1204	Op1205	Op1206	Op1207	Op1208	Op1209	Op1210	Op1211	Op1212	Op1213	Op1214	Op1215	Op1216	Op1217	Op1218	Op1219	Op1220	Op1221	Op1222	Op1223	Op1224	Op1225	Op1226	Op1227	Op1228	Op1229	Op1230	Op1231
---------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------



0C	00000000G	00	00000000G	59	DD	000A5	PUSHL	R9	...		
	0000G	CF		01	DD	000A7	PUSHL	#1	...		
				8F	DD	000A9	PUSHL	#LIB\$ INITERR	...		
				04	FB	000AF	CALLS	#4, LIB\$SIGNAL	...		
				03	E0	000B6	BBS	#3, LIB\$GL_CTLMSK+4, 7\$	...	1302	
			008C	C6	D5	000BC	TSTL	140(HEADER)	...	1303	
				06	13	000C0	BEQL	7\$	...		
	0000G	CF	80	8F	88	000C2	BISB2	#128, LIB\$GL_CRE8FLAGS	...	1305	
			0000G	CF	95	000C8	TSTB	LIB\$GL_CRE8FLAGS	...	1309	
				12	18	000CC	BGEQ	8\$	...		
			0000'	CF	9F	000CE	PUSHAB	DCX_MAP_DESC	...	1311	
			0000G	CF	9F	000D2	PUSHAB	LIB\$GL_CIBCTL	...		
	00000000G	00		02	FB	000D6	CALLS	#2, LBR\$DCX_MAP	...		
		48		50	E9	000DD	BLBC	STATUS, 12\$	...		
		50	0000G	CF	D0	000E0	MOVL	LIB\$GL_LIBFDB, R0	...	1313	
		57	0000G	CF	D0	000E5	MOVL	LIB\$GL_OUTFDB, R7	...	1314	
08	A7	08		08	28	000EA	MOVC3	#8, 8(R0), 8(R7)	...		
		10		5A	D0	000F0	MOVL	R10, 16(R8)	...	1315	
				50	CF	D0	000F4	MOVL	LIB\$GL_TYPE, R0	1319	
				52	0000G	CF	D0	000F9	MOVL	LIB\$AL_HDRLLEN[R0], USRMODHDRLEN	
					0000G	CF	95	000FF	TSTB	LIB\$GL_CTLMSK+3	1321
					07	18	00103	BGEQ	9\$		
	0000G	CF	40	A6	9A	00105	MOVZBL	60(HEADER), LIB\$AL_HDRLLEN[R0]	...	1323	
					0000G	CF	95	0010C	TSTB	LIB\$GL_CRE8FLAGS	1329
					09	18	00110	BGEQ	10\$		
			50	0000'	CF	9E	00112	MOVAB	DCX_MAP_DESC, R0		
					50	DD	00117	PUSHL	R0		
					02	11	00119	BRB	11\$		
					7E	D4	0011B	CLRL	-(SP)		
			0000'	CF	9F	0011D	PUSHAB	OUTLIBINDEX	...		
				57	DD	00121	PUSHL	R7	...		
	0000G	CF		03	FB	00123	CALLS	#3, LIB_CREATE_LIB	...		
		01		50	E8	00128	BLBS	STATUS, -13\$	...		
					04	0012B	RET		...		
					CF	D0	0012C	MOVL	LIB\$GL_TYPE, R0	1330	
	0000G	CF	40	52	D0	00131	MOVL	USRMODHDRLEN, LIB\$AL_HDRLLEN[R0]	...		
			13	0000G	CF	E9	00137	BLBC	LIB\$GL_CTLMSK+3, 14\$	1332	
				0A00	8F	BB	0013C	PUSHR	#^M<R9,R11>	1333	
					02	DD	00140	PUSHL	#2		
	00000000G	00	00000000G	8F	DD	00142	PUSHL	#LIB\$ CNVRTING	...		
		CF		04	FB	00148	CALLS	#4, LIB\$SIGNAL	...		
		19		00	FB	0014F	CALLS	#0, GET_INDEX_IF_NOT_EMPTY	...	1339	
				50	E8	00154	BLBS	STATUS, -15\$	...		
			00000000G	00	DD	00157	PUSHL	LBR\$GL_RMSSTV	...		
				50	DD	0015D	PUSHL	STATUS	...		
				5B	DD	0015F	PUSHL	R11	...		
				01	DD	00161	PUSHL	#1	...		
	00000000G	00	00000000G	8F	DD	00163	PUSHL	#LIB\$ INDEXERR	...		
				05	FB	00169	CALLS	#5, LIB\$SIGNAL	...		
				0000G	CF	95	00170	TSTB	LIB\$GL_CTLMSK+3	1340	
				25	18	00174	BGEQ	16\$	...		
				0000V	CF	9F	00176	PUSHAB	LIB_PUT_HISTORY	1343	
			0000G	CF	9F	0017A	PUSHAB	LIB\$GL_CIBCTL	...		
	00000000G	00		02	FB	0017E	CALLS	#2, LBR\$GET_HISTORY	...		
		13		50	E8	00185	BLBS	STATUS, 16\$	...	1344	
				50	DD	00188	PUSHL	STATUS	...	1346	
				5B	DD	0018A	PUSHL	R11	...		

00000000G	00	00000000G	01	DD	0018C	PUSHL	#1	:	
			8F	DD	0018E	PUSHL	#LIB\$ HISTERR	:	
		0000'	04	FB	00194	CALLS	#4, LIB\$SIGNAL	:	
00000000G	00		CF	9F	0019B	PUSHAB	OUTLIBINDEX	:	1350
	19		01	FB	0019F	CALLS	#1, LBR\$CLOSE	:	
		00000000G	50	E8	001A6	BLBS	STATUS, 17\$	:	
			00	DD	001A9	PUSHL	LBR\$GL_RMSSTV	:	
			50	DD	001AF	PUSHL	STATUS	:	
			59	DD	001B1	PUSHL	R9	:	
			01	DD	001B3	PUSHL	#1	:	
		00861058	8F	DD	001B5	PUSHL	#8786008	:	
00000000G	00		05	FB	001BB	CALLS	#5, LIB\$SIGNAL	:	
		0000G	CF	9F	001C2	PUSHAB	LIB\$GL_LIBCTL	:	1353
00000000G	00		01	FB	001C6	CALLS	#1, LBR\$CLOSE	:	
	19		50	E8	001CD	BLBS	STATUS, 18\$	:	
		00000000G	00	DD	001D0	PUSHL	LBR\$GL_RMSSTV	:	
			50	DD	001D6	PUSHL	STATUS	:	
			59	DD	001D8	PUSHL	R9	:	
			01	DD	001DA	PUSHL	#1	:	
		00861050	8F	DD	001DC	PUSHL	#8786000	:	
00000000G	00		05	FB	001E2	CALLS	#5, LIB\$SIGNAL	:	
	CF	80	8F	8A	001E9	BICB2	#128, LIB\$GL_CTLMSK+2	:	1355
	CF	0000G	CF	D0	001EF	MOVL	LIB\$GL_OUTFDB, LIB\$GL_LIBFDB	:	1356
			58	DD	001F6	PUSHL	R8	:	1360
		0000G	CF	9F	001F8	PUSHAB	LIB\$GL TYPE	:	
		04	AC	9F	001FC	PUSHAB	AFTER_FUNC	:	
		0000G	CF	9F	001FF	PUSHAB	LIB\$GL_LIBCTL	:	
00000000G	00		04	FB	00203	CALLS	#4, LBR\$INI_CONTROL	:	
	13		50	E8	0020A	BLBS	STATUS, 19\$	:	
			50	DD	0020D	PUSHL	STATUS	:	
			59	DD	0020F	PUSHL	R9	:	
			01	DD	00211	PUSHL	#1	:	
		00000000G	8F	DD	00213	PUSHL	#LIB\$ INITERR	:	
00000000G	00		04	FB	00219	CALLS	#4, LIB\$SIGNAL	:	
		0000G	CF	9F	00220	PUSHAB	LIB\$GL_LIBCTL	:	1363
00000000G	00		01	FB	00224	CALLS	#1, LBR\$OPEN	:	
	19		50	E8	0022B	BLBS	STATUS, 20\$	:	
		00000000G	00	DD	0022E	PUSHL	LBR\$GL_RMSSTV	:	
			50	DD	00234	PUSHL	STATUS	:	
			59	DD	00236	PUSHL	R9	:	
			01	DD	00238	PUSHL	#1	:	
		00861098	8F	DD	0023A	PUSHL	#8786072	:	
00000000G	00		05	FB	00240	CALLS	#5, LIB\$SIGNAL	:	
	50		01	D0	00247	MOVL	#1, R0	:	1365
			04	0024A	RET			:	1366

; Routine Size: 587 bytes, Routine Base: \$CODE\$ + 0000



```

: 310      1367 1 ROUTINE get_index_if_not_empty =
: 311      1368 2 BEGIN
: 312      1369 2 LOCAL
: 313      1370 2     status;
: 314      1371 2     |
: 315      1372 2     | Call the library procedures to return each entry in the module name
: 316      1373 2     | index. It will call copymodule for each entry. Treat the empty library
: 317      1374 2     | case benignly.
: 318      1375 2     |
: 319      1376 2     status = lbr$get_index (lib$gl_libctl, lib$gl_modnamix,      !Return the index
: 320      1377 2     copymodule);                                           !and call copymodule for each entry
: 321      1378 2
: 322      1379 2 IF .status EQL lbr$_nulidx THEN
: 323      1380 2     BEGIN
: 324      1381 2     signal (lib$_emptylibrary, 1, lib$gl_libfdb[fdb$l_namdesc]);
: 325      1382 2     status = ss$_normal;
: 326      1383 2     END;
: 327      1384 2
: 328      1385 2 RETURN .status;
: 329      1386 1 END;
```

0004 00000 GET_INDEX IF NOT EMPTY:						
				.WORD	Save R2	: 1367
	0000V	CF	9F 00002	PUSHAB	COPYMODULE	: 1376
	0000G	CF	9F 00006	PUSHAB	LIB\$GL_MODNAMIX	
	0000G	CF	9F 0000A	PUSHAB	LIB\$GL_LIBCTL	
	00000000G	00	03 FB 0000E	CALLS	#3, LBR\$GET_INDEX	
		52	50 D0 00015	MOVL	R0, STATUS	
	00000000G	8F	52 D1 00018	CMPL	STATUS, #LBR\$_NULIDX	: 1379
			18 12 0001F	BNEQ	1\$	
7E	0000G	CF	10 C1 00021	ADDL3	#16, LIB\$GL_LIBFDB, -(SP)	: 1381
			01 DD 00027	PUSHL	#1	
		00000000G	8F DD 00029	PUSHL	#LIB\$ EMPTYLIBRARY	
	00000000G	00	03 FB 0002F	CALLS	#3, LIB\$SIGNAL	
		52	01 D0 00036	MOVL	#1, STATUS	: 1382
		50	52 D0 00039 1\$:	MOVL	STATUS, R0	: 1385
			04 0003C	RET		: 1386

; Routine Size: 61 bytes, Routine Base: \$CODE\$ + 024B

```
1387 1 ROUTINE copymodule (keydesc, modrfa) =
1388 2 BEGIN
1389 2
1390 2 ++
1391 2 This routine is called by the librarian for each name in the
1392 2 module name index. The text for the name is read and inserted
1393 2 into the new library, and then the key is inserted into the
1394 2 index. If there is more than one index in the library, the other
1395 2 indices are searched to find all symbols associated with
1396 2 the module, and they are entered into the new library.
1397 2
1398 2 Inputs:
1399 2
1400 2     keydesc      address of string descriptor for module name
1401 2     modrfa       address of rfa of module
1402 2
1403 2 Outputs:
1404 2
1405 2     The module is copied into the output library
1406 2
1407 2 --
1408 2
1409 2 MAP
1410 2     keydesc : REF BBLOCK [dsc$c_s_bln];
1411 2
1412 2 LOCAL
1413 2     rms_status,           !Status from RMS operations
1414 2     first_put,           !flag true when first put done
1415 2     header : BBLOCK [lbr$c_pagesize], !Buffer for header
1416 2     bufdesc : BBLOCK [dsc$c_s_bln],   !descriptor for buffer
1417 2     rfa : BBLOCK [rfa$c_length];      !Dummy RFA
1418 2
1419 2 BIND
1420 2     libheader = .lbr$gl_control [lbr$l_hdrptr] : BBLOCK, !Point to the library header
1421 2     libdesc = lib$gl_libfdb [fdb$l_namdesc] : BBLOCK,   !Name the filename descriptor
1422 2     outdesc = lib$gl_outfdb [fdb$l_namdesc] : BBLOCK;   !...
1423 2
1424 2 P rms_perform (lbr$find (lib$gl_libctl, .modrfa),           !Lookup key to find text
1425 2     lib$lookuper, .lbr$gl_rmsstv, 2, .keydesc, libdesc);
1426 2
1427 2 bufdesc [dsc$a_pointer] = header;
1428 2 first_put = true;
1429 2
1430 2 !
1431 2 ! Read all text records for the module until end of file is returned. Write the records
1432 2 ! into the new library.
1433 2
1434 2 WHILE (bufdesc [dsc$w_length] = lbr$c_pagesize;
1435 2     rms_status = lbr$get_record (lib$gl_libctl, bufdesc, bufdesc); !Read all records of module
1436 2     IF NOT .rms_status AND (.rms_status NEQ rms$_eof)
1437 2     THEN BEGIN
1438 2         SIGNAL (lib$readerr, 1, libdesc, .rms_status, .lbr$gl_rmsstv);
1439 2         EXITLOOP;
1440 2     END;
1441 2
1442 2 .rms_status NEQ rms$_eof)
1443 2 DO BEGIN
```



```
388      1444 3 LOCAL
389      1445 3     status;
390      1446 3     status = lbr$put_record (outlibindex, bufdesc,      ! and write them to new library
391      1447 3         (IF .first_put THEN newtxtrfa ELSE rfa));
392      1448 3
393      1449 3     IF NOT .status
394      1450 3     THEN BEGIN      ! exit and write end of module record
395      1451 3         signal(lib$writeerr, 1, outdesc, .status, .lbr$gl_rmsstv);
396      1452 3         EXITLOOP;
397      1453 3     END;
398      1454 3
399      1455 3     first_put = false;
400      1456 3     END;
401      1457 3
402      1458 3     ! Text for module has been copied. Write end of module record
403      1459 3     rms_perform (lbr$put_end (outlibindex),      !Terminate PUT
404      1460 3         lib$writeerr, .lbr$gl_rmsstv, 1, outdesc);
405      1461 3
406      1462 3     ! Insert the module name into the new library
407      1463 3
408      1464 3     perform (lbr$set_index (outlibindex, lib$gl_modnamix), !Insert into module name index
409      1465 3         lib$_indexerr, 1, outdesc);
410      1466 3
411      1467 3     rms_perform (lbr$insert_key (outlibindex, .keydesc, newtxtrfa), !Insert key into index
412      1468 3         lib$_inserterr, .lbr$gl_rmsstv, 2, .keydesc, outdesc);
413      1469 3
414      1470 3
415      1471 3     ! Read module header from old library
416      1472 3
417      1473 3     bufdesc [dsc$w_length] = lbr$c_maxhdrsiz;
418      1474 3     bufdesc [dsc$a_pointer] = header;
419      1475 3     rms_perform (lbr$set_module (lib$gl_libctl, .modrfa, bufdesc, bufdesc),
420      1476 3         lib$_mhderr, .lbr$gl_rmsstv, 2, .keydesc, libdesc);
421      1477 3
422      1478 3     ! Set insert date/time of module in new library
423      1479 3
424      1480 3     lbr$insert_time (outlibindex, newtxtrfa, header [mhd$b_datim]);
425      1481 3     perform (lbr$set_index (lib$gl_libctl, lib$gl_modnamix),      !Set to old library
426      1482 3         lib$_indexerr, 1, libdesc);
427      1483 3
428      1484 3     ! If there is user information in the module header, update the module header
429      1485 3     ! in the new library.
430      1486 3
431      1487 3     IF .libheader [lhd$b_mhdusz] NEQ 0
432      1488 3     THEN BEGIN
433      1489 3         bufdesc [dsc$w_length] = .libheader [lhd$b_mhdusz];      !Set length of update data
434      1490 3         bufdesc [dsc$a_pointer] = header [mhd$b_usrdat];      !Point to update data
435      1491 3         rms_perform (lbr$set_module (outlibindex, newtxtrfa, 0, 0, bufdesc),      !Update module header
436      1492 3             lib$_mhderr, .lbr$gl_rmsstv, 2, .keydesc, outdesc);
437      1493 3         perform (lbr$set_index (lib$gl_libctl, lib$gl_modnamix),      !Set to old library
438      1494 3             lib$_indexerr, 1, libdesc);
439      1495 3     END;
440      1496 3
441      1497 3     ! If there are global symbols in the module, then search the index of the old library for them
442      1498 3     ! so they can be entered into the new library global symbol index
443      1499 3
444      1500 2 IF .libheader [lhd$b_nindex] GTR 1      !If there is more than one index
```

```

: 445      1501 3 THEN BEGIN
: 446      1502 3   INCRU i FROM 2 TO .libheader [lhd$b_nindex]           !Loop through the other indices
: 447      1503 4   DO BEGIN
: 448      1504 4       curindex = .i;                                     !Set current index number
: 449      1505 4       rms_perform (lbr$search (lib$gl_libctl,           !Search index for symbols
: 450      P 1506 4         curindex, .modrfa, enterglobals),             !so they can be entered in new library
: 451      1507 4         lib$_indexerr, .lbr$gl_rmsstv, 1, libdesc);
: 452      1508 4
: 453      1509 3   END;
: 454      1510 2   END;
: 455      1511 2
: 456      P 1512 2 perform (lbr$set_index (lib$gl_libctl, lib$gl_modnamix), !Do set index to set index number and lbr$gl_control
: 457      1513 2         lib$_indexerr, 1, libdesc);
: 458      1514 2
: 459      1515 2 lib_log_op (lib$_inserted, .keydesc, .lib$gl_outfdb); !Log on console if logging
: 460      1516 2
: 461      1517 2 RETURN true
: 462      1518 1 END;
                                     !Of copymodule
```

## OFFC 00000 COPYMODULE:

	5B	00000000G	00	9E	00002	.WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11	: 1387
	5A	00000000G	8F	D0	00009	MOVAB	LBR\$SET_INDEX, R11	
	59	00000000G	CF	9E	00010	MOVL	#LIB\$ INDEXERR, R10	
	58	00000000G	00	9E	00015	MOVAB	OUTLIBINDEX, R9	
	57	00000000G	00	9E	0001C	MOVAB	LBR\$GL_RMSSTV, R8	
	5E	FDF0	CE	9E	00023	MOVAB	LIB\$SIGNAL, R7	
	50	00000000G	00	D0	00028	MOVAB	-528(SP), \$P	
	52	0A	A0	D0	0002F	MOVL	LBR\$GL_CONTROL, R0	: 1420
55	0000G	CF	10	C1	00033	MOVL	10(R0), R2	
56	0000G	CF	10	C1	00039	ADDL3	#16, LIB\$GL_LIBFDB, R5	: 1421
		08	AC	DD	0003F	ADDL3	#16, LIB\$GL_OUTFDB, R6	: 1422
		0000G	CF	9F	00042	PUSHL	MODRFA	: 1425
	00000000G	00	02	FB	00046	PUSHAB	LIB\$GL_LIBCTL	
		14	50	E8	0004D	CALLS	#2, LBR\$FIND	
			68	DD	00050	BLBS	STATUS, 1\$	
			50	DD	00052	PUSHL	LBR\$GL_RMSSTV	
			55	DD	00054	PUSHL	STATUS	
		04	AC	DD	00056	PUSHL	R5	
			02	DD	00059	PUSHL	KEYDESC	
		00000000G	8F	DD	0005B	PUSHL	#2	
	67		06	FB	00061	PUSHL	#LIB\$ LOOKUPERR	
0C	AE	10	AE	9E	00064	CALLS	#6, LIB\$SIGNAL	: 1427
	54		01	D0	00069	MOVAB	HEADER, BUFDESC+4	: 1428
08	AE	0200	8F	B0	0006C	MOVL	#1, FIRST PUT	: 1434
		08	AE	9F	00072	MOVW	#512, BUFDESC	: 1435
		0C	AE	9F	00075	PUSHAB	BUFDESC	
		0000G	CF	9F	00078	PUSHAB	BUFDESC	
	00000000G	00	03	FB	0007C	PUSHAB	LIB\$GL_LIBCTL	
		53	50	D0	00083	CALLS	#3, LBR\$GET_RECORD	
		19	53	E8	00086	MOVL	R0, RMS STATUS	
0001827A	8F		53	D1	00089	BLBS	RMS STATUS, 3\$	: 1436
			10	13	00090	CMPL	RMS STATUS, #98938	
						BEQL	3\$	



		68	DD	00092	PUSHL	LBR\$GL_RMSSTV	: 1438
		53	DD	00094	PUSHL	RMS_STATUS	
		55	DD	00096	PUSHL	R5	
		01	DD	00098	PUSHL	#1	
	008610B2	8F	DD	0009A	PUSHL	#8786098	
		34	11	000A0	BRB	6\$	
0001827A	8F	53	D1	000A2	3\$: CMPL	RMS_STATUS, #98938	1442
		34	13	000A9	BEQL	8\$	
	06	54	E9	000AB	BLBC	FIRST_PUT, 4\$	1447
	50	A9	9E	000AE	MOVAB	NEWTXTRFA, R0	
		03	11	000B2	BRB	5\$	
	50	6E	9E	000B4	4\$: MOVAB	RFA, R0	
		50	DD	000B7	5\$: PUSHL	R0	
		AE	9F	000B9	PUSHAB	BUFDESC	1446
		59	DD	000BC	PUSHL	R9	
00000000G	00	03	FB	000BE	CALLS	#3, LBR\$PUT_RECORD	
	13	50	E8	000C5	BLBS	STATUS, 7\$	1448
		68	DD	000C8	PUSHL	LBR\$GL_RMSSTV	1450
		50	DD	000CA	PUSHL	STATUS	
		56	DD	000CC	PUSHL	R6	
		01	DD	000CE	PUSHL	#1	
	008610D2	8F	DD	000D0	PUSHL	#8786130	
	67	05	FB	000D6	6\$: CALLS	#5, LIB\$SIGNAL	1449
		04	11	000D9	BRB	8\$	
		54	D4	000DB	7\$: CLRL	FIRST_PUT	1454
		8D	11	000DD	BRB	2\$	1434
		59	DD	000DF	8\$: PUSHL	R9	1460
00000000G	00	01	FB	000E1	CALLS	#1, LBR\$PUT_END	
	11	50	E8	000E8	BLBS	STATUS, 9\$	
		68	DD	000EB	PUSHL	LBR\$GL_RMSSTV	
		50	DD	000ED	PUSHL	STATUS	
		56	DD	000EF	PUSHL	R6	
		01	DD	000F1	PUSHL	#1	
	008610D2	8F	DD	000F3	PUSHL	#8786130	
	67	05	FB	000F9	CALLS	#5, LIB\$SIGNAL	
		CF	9F	000FC	9\$: PUSHAB	LIB\$GL_MODNAMIX	1465
		59	DD	00100	PUSHL	R9	
		02	FB	00102	CALLS	#2, LBR\$SET_INDEX	
	6B	50	E8	00105	BLBS	STATUS, 10\$	
	0B	50	DD	00108	PUSHL	STATUS	
		56	DD	0010A	PUSHL	R6	
		01	DD	0010C	PUSHL	#1	
		5A	DD	0010E	PUSHL	R10	
	67	04	FB	00110	CALLS	#4, LIB\$SIGNAL	
		A9	9F	00113	10\$: PUSHAB	NEWTXTRFA	1468
	53	AC	D0	00116	MOVL	KEYDESC, R3	
		53	DD	0011A	PUSHL	R3	
		59	DD	0011C	PUSHL	R9	
00000000G	00	03	FB	0011E	CALLS	#3, LBR\$INSERT_KEY	
	13	50	E8	00125	BLBS	STATUS, 11\$	
		68	DD	00128	PUSHL	LBR\$GL_RMSSTV	
		50	DD	0012A	PUSHL	STATUS	
		8F	BB	0012C	PUSHR	#^M<R3,R6>	
		02	DD	00130	PUSHL	#2	
	00000000G	8F	DD	00132	PUSHL	#LIB\$ INSERTERR	
		06	FB	00138	CALLS	#6, LIB\$SIGNAL	
08	AE	80	8F	9B	0013B	11\$: MOVZBW	#128, BUFDESC : 1473

OC	AE	10	AE	9E	00140	MOVAB	HEADER, BUFDESC+4	1474
		08	AE	9F	00145	PUSHAB	BUFDESC	1476
		OC	AE	9F	00148	PUSHAB	BUFDESC	
		08	AC	DD	0014B	PUSHL	MODRFA	
00000000G	00	0000G	CF	9F	0014E	PUSHAB	LIB\$GL_LIBCTL	
	11		04	FB	00152	CALLS	#4, LBR\$SET_MODULE	
			50	E8	00159	BLBS	STATUS, 12\$	
			68	DD	0015C	PUSHL	LBR\$GL_RMSSTV	
			50	DD	0015E	PUSHL	STATUS	
			28	BB	00160	PUSHR	#^M<R3,R5>	
			02	DD	00162	PUSHL	#2	
	67	00000000G	8F	DD	00164	PUSHL	#LIB\$ MHDERR	
		18	06	FB	0016A	CALLS	#6, LIB\$SIGNAL	1480
		F8	AE	9F	0016D	PUSHAB	HEADER+8	
			A9	9F	00170	PUSHAB	NEWTXTRFA	
00000000G	00		59	DD	00173	PUSHL	R9	
		0000G	03	FB	00175	CALLS	#3, LBR\$INSERT TIME	
		0000G	CF	9F	0017C	PUSHAB	LIB\$GL_MODNAMIX	1482
	6B		CF	9F	00180	PUSHAB	LIB\$GL_LIBCTL	
	0B		02	FB	00184	CALLS	#2, LBR\$SET_INDEX	
			50	E8	00187	BLBS	STATUS, 13\$	
			50	DD	0018A	PUSHL	STATUS	
			55	DD	0018C	PUSHL	R5	
			01	DD	0018E	PUSHL	#1	
	67		5A	DD	00190	PUSHL	R10	
		3C	04	FB	00192	CALLS	#4, LIB\$SIGNAL	
			A2	95	00195	TSTB	60(R2)	1487
			4A	13	00198	BEQL	15\$	
08	AE	3C	A2	9B	0019A	MOVZBW	60(R2), BUFDESC	1489
OC	AE	20	AE	9E	0019F	MOVAB	HEADER+16, BUFDESC+4	1490
		08	AE	9F	001A4	PUSHAB	BUFDESC	1492
			7E	7C	001A7	CLRQ	-(SP)	
		F8	A9	9F	001A9	PUSHAB	NEWTXTRFA	
			59	DD	001AC	PUSHL	R9	
00000000G	00		05	FB	001AE	CALLS	#5, LBR\$SET_MODULE	
	13		50	E8	001B5	BLBS	STATUS, 14\$	
			68	DD	001B8	PUSHL	LBR\$GL_RMSSTV	
		0048	50	DD	001BA	PUSHL	STATUS	
			8F	BB	001BC	PUSHR	#^M<R3,R6>	
			02	DD	001C0	PUSHL	#2	
	67	00000000G	8F	DD	001C2	PUSHL	#LIB\$ MHDERR	
		0000G	06	FB	001C8	CALLS	#6, LIB\$SIGNAL	
		0000G	CF	9F	001CB	PUSHAB	LIB\$GL_MODNAMIX	1494
			CF	9F	001CF	PUSHAB	LIB\$GL_LIBCTL	
	6B		02	FB	001D3	CALLS	#2, LBR\$SET_INDEX	
	0B		50	E8	001D6	BLBS	STATUS, 15\$	
			50	DD	001D9	PUSHL	STATUS	
			55	DD	001DB	PUSHL	R5	
			01	DD	001DD	PUSHL	#1	
			5A	DD	001DF	PUSHL	R10	
	67		04	FB	001E1	CALLS	#4, LIB\$SIGNAL	
	52	01	A2	9A	001E4	MOVZBL	1(R2), R2	1500
	01		52	91	001E8	CMPB	R2, #1	
			35	1B	001EB	BLEQU	19\$	
	54		02	D0	001ED	MOVL	#2, I	1502
			2B	11	001F0	BRB	18\$	
F4	A9		54	D0	001F2	MOVL	I, CURINDEX	1504



		0000V	CF	9F	001F6	PUSHAB	ENTERGLOBALS		1507
		08	AC	DD	001FA	PUSHL	MODRFA		
		F4	A9	9F	001FD	PUSHAB	CURINDEX		
		0000G	CF	9F	00200	PUSHAB	LIB\$GL_LIBCTL		
00000000G	00		04	FB	00204	CALLS	#4, LBR\$SEARCH		
	0D		50	E8	0020B	BLBS	STATUS, 17\$		
			68	DD	0020E	PUSHL	LBR\$GL_RMSSTV		
			50	DD	00210	PUSHL	STATUS		
			55	DD	00212	PUSHL	R5		
			01	DD	00214	PUSHL	#1		
			5A	DD	00216	PUSHL	R10		
	67		05	FB	00218	CALLS	#5, LIB\$SIGNAL		
	52		54	D6	0021B	INCL	I		1502
			54	D1	0021D	CMPL	I, R2		
			D0	1B	00220	BLEQU	16\$		
		0000G	CF	9F	00222	PUSHAB	LIB\$GL_MODNAMIX		1513
		0000G	CF	9F	00226	PUSHAB	LIB\$GL_LIBCTL		
	6B		02	FB	0022A	CALLS	#2, LBR\$SET_INDEX		
	0B		50	E8	0022D	BLBS	STATUS, 20\$		
			50	DD	00230	PUSHL	STATUS		
			55	DD	00232	PUSHL	R5		
			01	DD	00234	PUSHL	#1		
			5A	DD	00236	PUSHL	R10		
	67		04	FB	00238	CALLS	#4, LIB\$SIGNAL		
		0000G	CF	DD	0023B	PUSHL	LIB\$GL_OUTFDB		1515
			53	DD	0023F	PUSHL	R3		
		00000000G	8F	DD	00241	PUSHL	#LIB\$ INSERTED		
0000G	CF		03	FB	00247	CALLS	#3, LIB_LOG_OP		
	50		01	D0	0024C	MOVL	#1, R0		1517
			04	0024F	RET				1518

; Routine Size: 592 bytes, Routine Base: \$CODE\$ + 0288

```
: 463      1519  1
: 464      1520  1 ROUTINE lib_put_history (rec_desc) =
: 465      1521  2 BEGIN
: 466      1522  2 !++
: 467      1523  2 !
: 468      1524  2 !--
: 469      1525  2 RETURN lbr$put_history ( outlibindex, .rec_desc );
: 470      1526  1 END;      ! of lib_put_history
```

		0000	00000	LIB_PUT_HISTORY:					
		04	AC	DD	00002	.WORD	Save nothing		1520
		0000	CF	9F	00005	PUSHL	REC_DESC		1525
00000000G	00		02	FB	00009	PUSHAB	OUTLIBINDEX		
			04	00010	CALLS	#2, LBR\$PUT_HISTORY			
					RET				1526

; Routine Size: 17 bytes, Routine Base: \$CODE\$ + 04D8

```

472 1527 1 ROUTINE enterglobals (keydesc) =
473 1528 2 BEGIN
474 1529 2 ++
475 1530 2
476 1531 2 This routine is called to enter a global symbol into the global symbol
477 1532 2 index for an object module
478 1533 2
479 1534 2 Inputs:
480 1535 2
481 1536 2 keydesc address of descriptor for symbol name
482 1537 2
483 1538 2 Outputs:
484 1539 2
485 1540 2 Global symbol name is entered into index of new library
486 1541 2
487 1542 2 --
488 1543 2
489 1544 2 MAP
490 1545 2 keydesc : REF BBLOCK; !Really a string descriptor
491 1546 2 BIND
492 1547 2 libdesc = lib$gl_libfdb [fdb$l_namdesc] : BBLOCK, !Name the filename descriptor
493 1548 2 outdesc = lib$gl_outfdb [fdb$l_namdesc] : BBLOCK; !...
494 1549 2
495 P 1550 2 perform (lbr$set_index (outlibindex, curindex),
496 1551 2 lib$_indexerr, 1, outdesc);
497 1552 2
498 P 1553 2 rms_perform (lbr$insert_key (outlibindex, .keydesc, newtxtrfa),
499 1554 2 lib$_inserterr, .lbr$gl_rmsstv, 2, .keydesc, outdesc);
500 1555 2
501 P 1556 2 perform (lbr$set_index (lib$gl_libctl, lib$gl_modnamix),
502 1557 2 lib$_indexerr, 1, libdesc);
503 1558 2
504 1559 2 RETURN true
505 1560 1 END; !of enterglobals
```

```

                                00FC 00000 ENTERGLOBALS:
                                .WORD Save R2,R3,R4,R5,R6,R7
57 00000000G 8F D0 00002 MOVL #LIB$ INDEXERR, R7 : 1527
56 00000000G 00 9E 00009 MOVAB LBR$SET_INDEX, R6
55 0000' CF 9E 00010 MOVAB OUTLIBINDEX, R5
54 00000000G 00 9E 00015 MOVAB LIB$SIGNAL, R4
53 0000G CF 10 C1 0001C ADDL3 #16, LIB$GL_LIBFDB, R3 : 1547
52 0000G CF 10 C1 00022 ADDL3 #16, LIB$GL_OUTFDB, R2 : 1548
                                F4 A5 9F 00028 PUSHAB CURINDEX : 1551
                                55 DD 0002B PUSHL R5
                                66 02 FB 0002D CALLS #2, LBR$SET_INDEX
                                0B 50 E8 00030 BLBS STATUS, 1$
                                50 DD 00033 PUSHL STATUS
                                52 DD 00035 PUSHL R2
                                01 DD 00037 PUSHL #1
                                57 DD 00039 PUSHL R7
                                64 04 FB 0003B CALLS #4, LIB$SIGNAL
                                F8 A5 9F 0003E 1$: PUSHAB NEWTXTRFA : 1554
```



```
00000000G 00      04 AC DD 00041    PUSHL KEYDESC
18          55 DD 00044    PUSHL R5
00000000G 00      03 FB 00046    CALLS #3, LBR$INSERT_KEY
50          50 E8 0004D    BLBS STATUS, 2$
00000000G 00      00 DD 00050    PUSHL LBR$GL_RMSSTV
50          50 DD 00056    PUSHL STATUS
52          52 DD 00058    PUSHL R2
04          04 AC DD 0005A    PUSHL KEYDESC
02          02 DD 0005D    PUSHL #2
00000000G 8F DD 0005F    PUSHL #LIB$ INSERTERR
64          06 FB 00065    CALLS #6, LIB$SIGNAL
0000G      CF 9F 00068 2$:    PUSHAB LIB$GL_MODNAMIX
0000G      CF 9F 0006C    PUSHAB LIB$GL_LIBCTL
66          02 FB 00070    CALLS #2, LBR$SET_INDEX
0B          50 E8 00073    BLBS STATUS, 3$
50          50 DD 00076    PUSHL STATUS
53          53 DD 00078    PUSHL R3
01          01 DD 0007A    PUSHL #1
57          57 DD 0007C    PUSHL R7
64          04 FB 0007E    CALLS #4, LIB$SIGNAL
50          01 D0 00081 3$:    MOVL #1, R0
04          04 00084    RET
```

1557

1559  
1560

; Routine Size: 133 bytes, Routine Base: \$CODE\$ + 04E9

```
: 506      1561 1 END      ! Of module
: 507      1562 0 ELUDOM
```

.EXTRN LIB\$SIGNAL

## PSECT SUMMARY

Name	Bytes	Attributes
\$GLOBALS	8	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$OWNS	20	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$CODE\$	1390	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

## Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	32	0	581	00:01.1

LIB COMPRESS  
V04=000

G 10  
16-Sep-1984 01:46:57  
14-Sep-1984 12:37:58

VAX-11 Bliss-32 V4.0-742  
[LIBRAR.SRC]COMPRESS.B32;1

Page 20  
(6)

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:COMPRESS/OBJ=OBJ\$:COMPRESS MSRC\$:COMPRESS/UPDATE=(ENH\$:COMPRESS)

: Size: 1390 code + 28 data bytes  
: Run Time: 00:30.1  
: Elapsed Time: 00:59.8  
: Lines/CPU Min: 3117  
: Lexemes/CPU-Min: 34449  
: Memory Used: 262 pages  
: Compilation Complete



0200 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

TRANSFER  
LIS

DATABASE  
LIS

PUTCACHE  
LIS

LIBRAR

PREFIX  
REQ

LIBRARIAN  
MAP

CROSS  
LIS

SUBS  
LIS

PADLBR  
LIS

COMPRESS  
LIS

LIB  
MDL

FILEIO  
LIS

EXTRACT  
LIS

DELETE  
LIS